AF IFW

PTO/SB/17 (12-04v2)
Approved for use through 07/31/2006. OMB 0651-0032
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE
Under the Paperwork Reduction Act of 1995 no persons are required to respond to a collection of information unless it displays a valid OMB control number

Effective on 12/08/2004.
Fees pursuant to the Consolidated Appropriations Act, 2005 (H.R. 4818).

FEE TRANSMITTAL
for FY 2005

☐ Applicant claims small entity status. See 37 CFR 1.27

TOTAL AMOUNT OF PAYMENT ($) 500.00

Complete if Known	
Application Number	09/622,331
Filing Date	March 19, 2001
First Named Inventor	Mehmet Kemal Ozkan
Examiner Name	Hai V. Tran
Art Unit	2611
Attorney Docket No.	RCA 89,400

METHOD OF PAYMENT *(check all that apply)*

☐ Check ☐ Credit card ☐ Money Order ☐ None ☐ Other (please identify): ____________

Customer Number 24498

☒ Deposit Account: **Deposit Account Number** 07-0832 Deposit Account Name: **THOMSON LICENSING INC.**

For the above-identified deposit account, the Director is hereby authorized to: (check all that apply)

☒ Charge fee(s) indicated below
☐ Charge fee(s) indicated below, **except for the filing fee**
☒ Charge any additional fee(s) or underpayments of fee(s) under 37 CFR 1.16 and 1.17
☒ Credit any overpayments

WARNING: Information on this form may become public. Credit card information should not be included on this form. Provide credit card information and authorization on PTO-2038.

FEE CALCULATION

1. BASIC FILING, SEARCH, AND EXAMINATION FEES

	FILING FEES		SEARCH FEES		EXAMINATION FEES		
		Small Entity		Small Entity		Small Entity	
Application Type	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fee ($)	Fees Paid ($)
Utility	300	150	500	250	200	100	
Design	200	100	100	50	130	65	
Plant	200	100	300	150	160	80	
Reissue	300	150	500	250	600	300	
Provisional	200	100	0	0	0	0	

2. EXCESS CLAIM FEES

Fee Description	Fee ($)	Small Entity Fee ($)
Each claim over 20 (including Reissues)	50	25
Each independent claim over 3 (including Reissues)	200	100
Multiple dependent claims	360	180

Total Claims ________ - 20 or HP = **Extra Claims** ________ x **Fee ($)** $50 = **Fee Paid ($)** $ ________
HP = highest number of total claims paid for, if greater than 20.

Multiple Dependent Claims
Fee ($) ________ **Fee Paid ($)** ________

Independent Claims ________ - 3 or HP = **Extra Claims** 0 x **Fee ($)** $200 = **Fee Paid ($)** 0
HP = highest number of independent claims paid for, if greater than 3.

3. APPLICATION SIZE FEE

If the specification and drawings exceed 100 sheets of paper (excluding electronically filed sequence or computer listings under 37 CFR 1.52(e)), the application size fee due is $250 ($125 for small entity) for each additional 50 sheets or fraction thereof. See 35 U.S.C. 41(a)(1)(G) and 37 CFR 1.16(s).

Total Sheets ________ - 100 = **Extra Sheets** ________ / 50 = **Number of each additional 50 or fraction thereof** ________ (round **up** to a whole number) x **Fee ($)** ________ = **Fee Paid ($)** ________

4. OTHER FEE(S)

	Fees Paid ($)
Non-English Specification, $130 fee (no small entity discount)	
Other (e.g., late filing surcharge): **FEE FOR FILING APPEAL BRIEF - $500.00**	$500.00

SUBMITTED BY

Name (Print/Type)	JOSEPH J. OPALACH	*Registration No. (Attorney/Agent)*	36,229	*Telephone*	(609) 734-6839
Signature					March 22, 2006

This collection of information is required by 37 CFR 1.136. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.14. This collection is estimated to take 30 minutes to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. **SEND TO: Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.** If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Before the Board of Patent Appeals and Interferences

Applicants : Mehmet Kemal Ozkan et al.
Application No.: 09/622,331
Filed : March 19, 2001
For : A SYSTEM FOR FORMING, PARTITIONING AND PROCESSING PROGRAM GUIDES
Examiner : Hai V. Tran
Art Unit : 2611

APPEAL BRIEF

Mail Stop: Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

May It Please The Honorable Board:

Appellants appeal from the FINAL Office Action dated December 1, 2005, and the Advisory Action dated December 29, 2005, in which claims 1-16 of the above-identified application stand rejected.

Appellants waive an Oral Hearing for this appeal.

Please charge the $500.00 fee for filing this Brief to Deposit Account No. 07-0832.

Please charge any additional fee or credit overpayment to the above-indicated Deposit Account.

****

Certificate of Mailing under 37 CFR 1.8

I hereby certify that this correspondence is being deposited with the United States Postal Service as first class mail in a postage paid envelope addressed to: Mail Stop: Appeal Brief - Patents, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450 on the date indicated below.

Signature Patricia M. Fedorowycz Date: March 22, 2006
PATRICIA M. FEDOROWYCZ

TABLE OF CONTENTS

I. REAL PARTY IN INTEREST

The real party in interest of Application No. 09/622,331 is:

THOMSON LICENSING
46, Quai A. Le Gallo
F-92100 Boulogne-Billancourt
France.

II. RELATED APPEALS AND INTERFERENCES

There are no related Appeals or Interferences.

III. STATUS OF THE CLAIMS

Claims 1-16 are pending in this application. Claims 17 and 18 have been canceled.

Claims 1-16 have been rejected.

The rejection of claims 1-16 are appealed.

IV. STATUS OF AMENDMENTS

There have been no amendments after the FINAL Office Action dated December 1, 2005.

A response to the FINAL Office Action dated December 1, 2005, was filed by Appellants' representative on December 13, 2005 seeking reconsideration. An Advisory Action dated December 29, 2005 maintained the FINAL rejection, to which Appellants' representative filed a Notice of Appeal on February 9, 2006.

This appeal is directed to the claims as they stood at the time of the FINAL Office Action of December 1, 2005, which are shown in the Claims Appendix of this Brief.

V. SUMMARY OF CLAIMED SUBJECT MATTER

There are four independent claims in the application: 1, 6, 8 and 13.

Appellants' independent claims are directed to forming, partitioning and processing of program guides. (Appellants' specification, Title; and p. 2, lns. 24-34.) Program guides comprise a number of tables. (Appellants' specification, p. 2, lns. 24-34; FIGs. 5, 6, and 7.)

In this regard, independent claim 1 is directed to an apparatus that detects changes in tables of a program guide by the use of specific types of version information. In particular, claim 1 is directed to an apparatus that uses a first version identifier to determine if something has changed in the program guide and, if something has changed, uses a second version identifier to determine what particular table has changed. (Claim 1, lns. 14-16; Appellants' specification, p. 2, lns. 26-29; p. 14, ln. 32 to p. 15, ln. 28; FIGs. 5, 6 and 7.)

Turning next to independent claim 6, this claim is directed to an apparatus that processes a program guide having individual partitions, where each partition is assigned to a partition identifier and wherein the partitions are dynamically re-partitionable by re-assignment of the partition identifiers. (Claim 6, lns. 3-8; p. 6, ln. 37 to p. 7, ln. 23; p. 11, lns. 20-29 and FIGs. 1, 9 and 10; p. 11, ln. 36 to p. 12, ln. 20 and FIGs. 4 and 8.)

Finally, independent claims 8 and 13 are complementary to claims 1 and 6 in that claims 8 and 13 are directed to methods of forming the packetized program data. In particular, claim 8 is directed to a method for forming packetized program data for processing in a decoder, where the packetized program data includes a first version identifier and a second version identifier. (Claim 8, lns. 5-12; Appellants' specification, p. 16, lns. 28-33; p. 17, lns. 18-20; FIG. 12.) With respect to claim 13, this claim is directed to a method for forming packetized program data for processing in a decoder, where the packetized program data includes updatable version numbers for indicating content change of a partition and cell numbers that enable dynamic re-partitioning of program guide information. (Claim 13, lns. 3-12; Appellants' specification, p. 16, ln. 28 to p. 17, ln. 29; FIG. 12.)

VI. GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1 to 16 have been rejected under 35 U.S.C. § 103(a) as being unpatentable over U.S. Patent No. 6,160,545 issued December 12, 2000 to Eyer et al. (*Eyer*) in view of the "Program Guide for Digital Television", ATSC Standard, Doc. A/55 (the *Guide*).

VII. ARGUMENT

Rejection of Claims 1-16 under 35 U.S.C. § 103(a) as being unpatentable over *Eyer* in view of the *Guide*

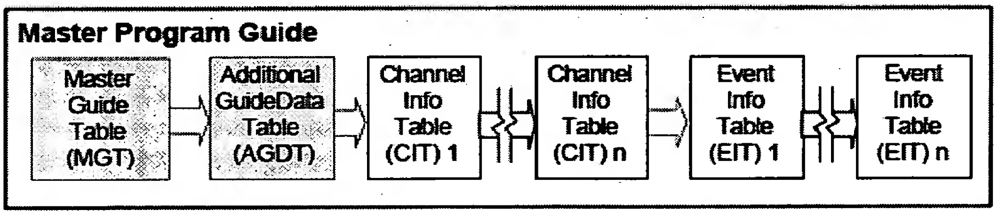
CLAIMS 1-5

The Examiner mischaracterizes the *Guide*. The plain meaning of the *Guide* supports Appellants' argument that the combination of *Eyer* in view of the *Guide* does not yield the requirements of Appellants' claims 1-5.

THE EXAMINER'S ANALYSIS OF THE *GUIDE* IS WRONG

The heart of the Examiner's argument is that the version numbers described in the *Guide* are the same as that required by Appellants' claim 1. Simply put, the Examiner is wrong.

Turning first to the *Guide*, a "Master Program Guide" (MPG) is defined as shown below.



The *Guide*, p. 5; Figure 5.1.

As can be observed from Figure 5.1 of the *Guide*, the MPG comprises the Master Guide Table (MGT), the Additional Guide Data Table (AGDT), Channel Information Tables (CITs) and Event Information Tables (EITs). In addition, the *Guide* defines a "version_number" parameter for each of these tables. (The *Guide*, p. 7, p. 13, p. 20 and p. 25.) In particular, in all of these tables, the definition of the "version number" is IDENTICAL and the version number represents the version number of the WHOLE program guide:

> version_number — This 5-bit field is the version number of the whole program guide. The version number *shall be incremented by 1 modulo 32 when a field in either the MPG or any SPG changes* with the exception of the life_time and actual_time fields in the Master Guide Table.
>
> The *Guide*, p. 8, 13, 21 and 27, emphasis added.

As a result, although there are a plurality of version numbers in the *Guide*, each of these version numbers operate in an IDENTICAL fashion, i.e., each of the version numbers track each other.

For example, in light of the *Guide* definition, if the value of the "version_number" parameter was initially "1" in the MGT, the value of the "version_number" parameter in the AGDT, CITs and EITs would be the same, i.e., equal to "1". This is required by the *Guide* definition since the value of the "version_number" parameter represents the version number of the whole program guide. Again, values for the "version_number" parameter in each table of the *Guide* **must track each other**. Indeed, the "version_number" parameter can not be different in each table of the MPG since, if any value was different, there would be a multiplicity of values and a resultant ambiguity as to the version number for the whole program guide. This would clearly be contrary to the "version_number" parameter definition found in the *Guide*.

In addition, the above-noted *Guide* definition also specifies how to change the value for the "version_number" parameter in all of the tables. In particular, if a field in the MPG changes (again, the MPG includes the MGT, AGDT, CIT and EIT tables), the *Guide* definition for the "version_number" in each of the tables requires that the value of the "version_number" parameter be incremented by 1 modulo 32. (The *Guide*, p. 8, 13, 21 and 27.) Continuing with the above example, if a field in an EIT of the MPG changes then the new value for the "version_number" parameter in is "2". Since — as required by the *Guide* definition — the "version_number" represents the version number of the whole program guide, the values for the "version_number" parameters in the Master Guide Table (MGT), the Additional Guide Data Table (AGDT), Channel Information Tables (CITs) and Event Information Tables (EITs) are all set equal to a value of "2". Thus, the values for the "version_number" parameters change in other tables even if the contents of those tables did not change. Again, all the "version_number" parameters in all the tables **track each other** — this is required by the *Guide* definition. (The *Guide*, p. 8, 13, 21 and 27.)

In view of the above, and as can be observed from the *Guide* definition, the "version_number" parameter value indicates whether a field in the MPG has changed — and the MPG includes the Master Guide Table (MGT), the Additional Guide Data Table (AGDT), Channel Information Tables (CITs) and Event Information Tables (EITs) as shown in Figure 5.1 of the *Guide*, above. As such, while the "version_number" parameter of the *Guide* indicates that a field has changed in the MPG— the "version_number" parameter value of the *Guide* does not indicate what field - or table - has changed in the MPG. In contrast, and as described further below, Appellants' claimed invention uses the first version identifier to determine if something has changed in the program guide and, if something has changed, uses the second version identifier to determine what particular table has changed. (Appellants' specification, p. 2, lns. 26-29.)

With the above description of the *Guide* as background, it can be shown that the Examiner mischaracterizes the *Guide*. For example, in the Advisory Action of 29th December 2005, the Examiner states:

> Off [sic] course, the version_number represents the whole Guide as Applicant indicated at the beginning when the GUIDE is first created; But after that, the change occurs within these tables because Provider changes their Guide content. Thus, if, for example, one of the table is changed then the corresponding version_number of the .corresponding Table_id is changed accordingly in which each corresponding table will be updated independently **(according to the well [sic] technique of "relational database and datastructure")**. The system will compare version_number between corresponding table, i.e., previous version_number value versus currently received updated table (same table_Id) with new version_number value. If there is a different then the corresponding table is updates [sic] accordingly.
>
> Advisory Action, p. 2, December 29, 2005; emphasis added.

The Examiner's statement that "each corresponding table [of the *Guide*] is updated independently" is wrong — and clearly conflicts with the *Guide* definition. Thus, the Examiner's following characterization "The system will compare version_number ..." is also wrong.

First, as noted earlier, the *Guide* definition states that the "version_number" parameter value in each table represents the version number of the whole program guide. (The *Guide*, p. 8, 13, 21 and 27.) As such, each corresponding table cannot be updated independently as asserted by the Examiner else there would be different version numbers in each table and, thus, no representation of the version number of the whole program guide as required by the *Guide*.

Second, as noted earlier, the *Guide* definition states that the "version_number" parameter value indicates whether a field in the MPG has changed — any field — in any table. Thus, each corresponding table cannot be updated independently as asserted by the Examiner because the "version_number" parameter as defined in the *Guide* for each table just indicates that "a" field in the whole MPG has changed — the version_number parameter as defined in the Guide does not indicate what field (or table) has changed.

In view of the above, there is simply no support in the *Guide* definition — let alone the *Guide* — for the Examiner's statement that "each corresponding table is updated independently."

In point of fact, the Examiner has found no support in the *Guide* for this statement since, instead, the Examiner states that such an operation is **according to the well [sic] technique of "relational database and datastructure"**. Yet this also fails for at least two reasons. First, whatever the Examiner's asserted "operation" is, the Examiner still completely disregards all the definitions of all the "version_number" parameters in all the tables of the *Guide*. The *Guide* simply does not operate in the way stated by the Examiner. And, second, the Examiner has provided no citation to prior art — before Appellants' filing date — that even defines or explains this so-called asserted **technique of "relational database and datastructure"**. Indeed, the Examiner appears to be performing an improper hindsight, or ex post facto, analysis and an incorrect analysis at that.

Therefore, although the *Guide* may describe a plurality of version numbers, the *Guide* does not describe or suggest a second version identifier and a processor for determining a change using a first version identifier and a second version identifier as required by Appellants' claim 1.

CLAIMS 1-5 ARE PATENTABLE

Attention is now directed to the requirements of Appellants' independent claim 1. For example, this claim requires:

> a processor for acquiring program guide information ...
>
> (a) a first version identifier conveyed in a primary data table and updated in response to a version change in at least one of a plurality of secondary tables hierarchically linked to said primary data table, and
>
> a processor for acquiring program guide information and
>
> (b) a second version identifier conveyed in a secondary data table and updated in response to at least one of, a version change in said secondary table, and a version change in a tertiary table hierarchically linked to said secondary table; and
>
> a processor for determining change in said secondary data table content by examining said second version identifier for a change **following determination of a change** in said first version identifier.

Claim 1, lns. 6-16; emphasis added.

In other words, Appellants' claimed invention uses the first version identifier to determine if something has changed in the program guide and, if something has changed, uses the second version identifier to determine what particular table has changed. (Appellants' specification, p. 2, lns. 26-29.)

In contrast, *Eyer* merely describes the use of a single version number for a block of data. (*Eyer*, col. 13, lns. 37-42.) Nowhere does *Eyer* describe or suggest a first version identifier, a second version identifier and a processor for determining a change as required by Appellants' claim 1.

Likewise, the *Guide* is also found lacking with respect to these requirements of Appellants' claim 1. In particular, although the *Guide* (described in more detail above) describes a plurality of version numbers so that, arguably, there is a first version identifier and a second version identifier — the *Guide* requires that all of these version numbers track each other. (The *Guide*, p. 8, 13, 21 and 27.) As such, a version number as defined in the *Guide* only indicates if there was a change in the program guide — not what has changed. Thus, nowhere does the *Guide* describes or suggest a second version identifier and a processor for determining a change using a first version identifier and a second version identifier as required by Appellants' claim 1.

Finally, it should be noted that both *Eyer* and the *Guide* teach away from Appellants' claimed invention. In particular, both *Eyer* and the *Guide* describe the use of a version number to track any change to the data by comparison to a previous version number. (*Eyer*, col. 20, lns. 8-18; the *Guide*, p. 8, 13, 21 and 27.) Nowhere does either *Eyer* or the *Guide*, singly or in combination, describe or suggest the use of a first version identifier to determine if something has changed in the program guide and, if something has changed, to examine a second version identifier to determine what particular table has changed as required by Appellants' claim 1.

Therefore, Appellants' claim 1, and dependant claims 2-5, are patentable since the combination of *Eyer* and the *Guide* does not yield the requirements of Appellants' claim 1.

CLAIMS 6-7, 13-16

CLAIMS 6-7 ARE PATENTABLE

The combination of *Eyer* in view of the *Guide* does not yield the requirements of Appellants' claims 6-7.

The Examiner states that:

> Eyer discloses an apparatus (Fig. 1 and 2) for acquiring packetized program data from at least a first source, in which Eyer apparatus acquires program guide information (IPG data) and for acquiring ancillary information conveyed in hierarchically ordered data tables in the packetized program data, the ancillary information including an initial master program guide with "block_version" is used to indicate change in programming has occurred during the valid lifetime of the current master program guide (Col. 13, lines 35-42+) and a processor for determining change and changes the program guide as needed.
>
> Advisory Action, p. 2, December 29, 2005.

For the sake of argument, Appellants agree.

Then the Examiner states:

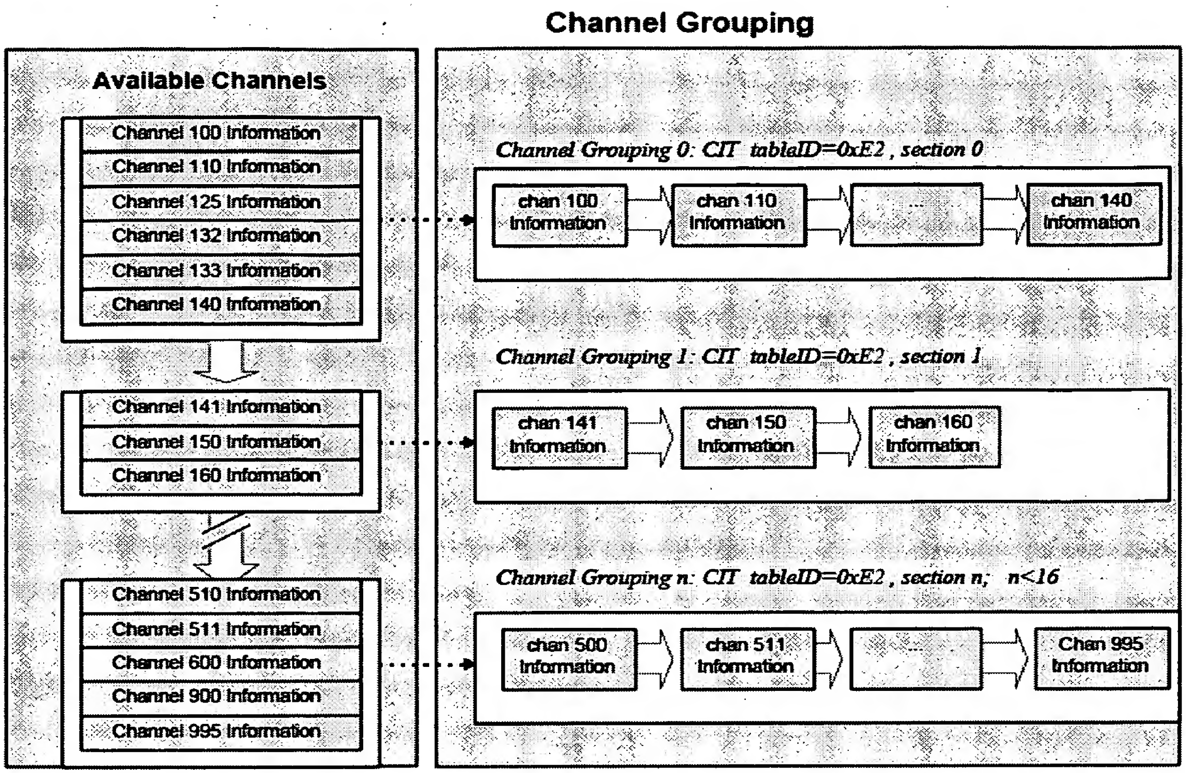
> In doing so, Eyer (Fig. 2-4) in view of ATSC discloses apparatus for adaptively decoding re-partitionable packetized program guide data according to region (see Eyer; Fig. 4) comprising a processor for acquiring program guide data comprising hierarchically ordered data table partitions and including partitioning information (Col. 8, lines 23-Col. 9, lines 15), in which the partitioning information including partition identifiers assigned to individual partition of said program guide data , as disclosed by ATSC standard page 19; Channel grouping, see Fig. 5.6 and EIT[sic]-EIT Link, Fig. 5.7 page 24.
>
> Advisory Action, p. 2, December 29, 2005; emphasis added.

The Examiner seems to be arguing that the Channel Grouping of the Channel Information Table (CIT) and/or the CIT-EIT Link of the *Guide* describe re-partitionable identifiers. The Examiner is wrong.

With respect to the channel groupings of the CIT table, the Guide states:

> [t]he channel information table may be divided into up to 16 sections (0 to 15), **each of which corresponds to one channel grouping** ...
>
> [t]he entire CIT table is simply a concatenation of all the sections (channel groupings) in order.
>
> The *Guide*, p. 18; emphasis added.

The Channel Groupings are also illustrated in Figure 5.6 of the Guide, shown below.



The *Guide*, p. 19; Figure 5.6.

It can be observed from Figure 5.6 of the *Guide* that Channel Grouping 0 includes channel 100, channel 110 etc. Since each Channel Grouping corresponds to a section of the CIT, it can be further observed that the corresponding section of the CIT includes a list of channel numbers. For example, for Channel Grouping 0 (section 0) this would include a literal list of channel numbers comprising channel 100, channel 110, etc.. This is further supported by reference to Table 5.7 on p. 20 of the *Guide*. Table 5.7 of the *Guide* clearly shows that for each section (Channel Grouping) identified by the "section_number" parameter there is literally a list of channel numbers as indicated by the "for loop" portion of Table 5.7 of the Guide. Further, as required by the *Guide*, each section of the CIT must occur in order, i.e., in sequence. (The *Guide*, p. 18.) In other words, section 0 (section_number = 0) must be followed by section 1 (section_number = 1).

As such, it could be argued that the Channel Groupings of the *Guide* at one time represents one partition of the channels and, at another time, the Channel Groupings of the *Guide* can be re-partitioned to represent a different channel grouping. But this is not what Appellants claim.

Appellants' independent claim 6 requires:

> a processor for acquiring program guide data comprising hierarchically ordered data table partitions and including partitioning information, said partitioning information including,
>
> partition identifiers assigned to individual partitions of said program guide data, wherein said program guide data partitions are dynamically re-partitionable **by re-assignment** of said partition identifiers in said partitioning information; and
>
> a processor for identifying said re-assigned partition identifiers and for acquiring additional program guide data in response to said identified re-assigned partition identifiers.
>
> Claim 8, lns. 5-12; emphasis added.

First, Appellants' independent claim 6 requires a "partition identifier". It should be noted that the Examiner has not actually stated what part of the *Guide* corresponds to Appellants' claimed "partition identifier". Regardless, Appellants will assume that one could argue that the "section_number" of Table 5.7 of the *Guide* that identifies the particular Channel Grouping is a partition identifier.

Second, Appellants claim 6 requires that partitions are dynamically re-partitionable **by re-assignment** of said partition identifiers in said partitioning information. This is not described or suggested in the *Guide*. While the contents themselves of a Channel Grouping can be changed over time, the value for the "section_number" cannot be re-assigned.

In other words, if a Channel Grouping or "section_number" of the *Guide* is a partition identifier, Appellants' claim 6 further requires that the value for the Channel Grouping or "section_number" of the Guide could be re-assigned from, e.g., a value of 0 to a value of 1. (For example, see Appellants' specification, p. 11, ln. 36 to p. 12, ln. 9.) Thus, the channels previously associated with channel grouping 0 could now be associated with channel grouping 1 by merely changing the value of the "section_number" parameter. However, and as noted above, while the contents, or list of channels, in a Channel Grouping of the *Guide* can be changed over time, the value for the "section_number" cannot be re-assigned. As stated in the Guide, **each section must occur in order**. (The *Guide*, p. 18.) As such, one cannot just change the value for section_number — it would now be out of order. Thus, the Channel Grouping described in the Guide does not describe "partition identifiers assigned to individual partitions of said program guide data, wherein said program guide data partitions are dynamically re-partitionable **by re-assignment** of said partition identifiers in said partitioning information" as required by Appellants' independent claim 6.

Similar comments apply to the CIT-EIT link of FIG. 5.7 on p. 24 of the *Guide*. In the Final Office Action dated December 1, 2005, the Examiner in particular pointed to the table_id extension of Table 5.8 of the *Guide* with respect to the CIT-EIT link. However, the table_id_extension of the *Guide* simply associates a given event information table (EIT) with a particular section and channel in that section. (The *Guide*, p. 26.) As such, the table_id_extension is not dynamically re-partitionable as required by Appellants' claim 6 since this is a part of the event information table that is fixed to a particular channel.

Finally, Appellants note that independent claim 6 requires a processor "for identifying the re-assigned partition identifiers and for acquiring additional program guide data." The Examiner still has not addressed this claim limitation.

Turning now to *Eyer*, this reference also does not describe a partition identifier as required by Appellants' claim 6. While it can be argued that *Eyer* describes partitioning of program guide data into global and regional data as illustrated in FIG. 4 of *Eyer*, nowhere does *Eyer* describe a partition identifier as claimed by Appellant.

Therefore, Appellants' independent claim 6, and dependant claim 7, are patentable since the combination of *Eyer* and the *Guide* does not yield the requirements of Appellants' claim 6.

CLAIMS 13-16 ARE PATENTABLE

Similar comments apply to Appellants independent claim 13. In particular, independent claim 13 requires:

> cell numbers assigned to individual partitions of said program guide information, wherein said program guide information cell partitions are dynamically re-partitionable by re-assignment of said cell number in said database;
>
> Claim 13, lns. 8-10; emphasis added.

As such, Appellants claims 13-16 stand or fall with Appellants independent claim 6.

CLAIMS 8-12

The Examiner mischaracterizes the *Guide* with respect to claim 8 in like fashion as described above with respect to claim 1. The plain meaning of the *Guide* supports Appellants' argument that the combination of *Eyer* in view of the *Guide* does not yield the requirements of Appellants' claims 8-12.

THE EXAMINER'S ANALYSIS OF THE *GUIDE* IS WRONG

The Examiner has applied the same analysis of the *Guide* to claim 8 as was applied to claim 1. As such, and for the same reasons as described above with respect to claim 1, the Examiner's analysis of the *Guide* as applied to claim 8 is also wrong.

Therefore, although the *Guide* may describe a plurality of version numbers, the *Guide* does not describe or suggest or a first version identifier and a second version identifier that are updated differently as required by Appellants' claim 8 (described below).

CLAIMS 8-12 ARE PATENTABLE

Appellants' independent claim 8 requires:

> (a) <u>a first version identifier</u> conveyed in a primary data table and <u>updated in response to a version change in at least one of a plurality of secondary tables hierarchically linked to said primary data table</u>, and
>
> (b) <u>a second version identifier</u> conveyed in a secondary data table and <u>updated in response to at least one of,</u>
>
> <u>a version change in said secondary table, and</u>
>
> <u>a version change in a tertiary table hierarchically linked to said secondary table</u>;
>
> Claim 8, lns. 5-12; emphasis added.

As can be observed from the above, Applicants' independent claim 8 requires a first version identifier and a second version identifier that are <u>updated differently</u>.

In contrast, *Eyer* merely describes the use of a single version number for a block of data. (*Eyer*, col. 13, lns. 37-42.) <u>Nowhere</u> does *Eyer* describe or suggest a first version identifier and a second version identifier <u>that are updated differently as required</u> by Appellants' claim 8.

In addition, and as noted earlier with respect to the argument for claim 1, the version numbers defined in the *Guide* represent the whole program guide and, as such, are <u>updated identically</u>. That is, the version numbers in the *Guide* <u>track each other</u>. Thus, the version numbers of the *Guide* <u>are not updated differently as required</u> by Appellants' independent claim 8.

Therefore, Appellants' independent claim 8, and dependant claims 9-12, are patentable since the combination of *Eyer* and the *Guide* does not yield the requirements of Appellants' claim 8.

VIII. CONCLUSION

For the above reasons, Appellants submit that claims 1-16 are patentable over *Eyer* in view of the *Guide*. It is therefore respectfully requested that the rejection of claims 1-16 under 35 U.S.C. § 103(a) be reversed.

Respectfully submitted,
Mehmet Kemal Ozkan et al.

By________________
Joseph J. Opalach
Registration No.: 36,229
(609) 734-6839

Patent Operations
Thomson Licensing, Inc.
P.O. Box 5312
Princeton, New Jersey 08543-5312

March 21, 2006

IX. CLAIMS APPENDIX

1. (Original) Apparatus for acquiring packetized program data from at least a first source, comprising:

a processor for acquiring program guide information and for acquiring ancillary information conveyed in hierarchically ordered data tables in said packetized program data, said ancillary information including,

(a) a first version identifier conveyed in a primary data table and updated in response to a version change in at least one of a plurality of secondary tables hierarchically linked to said primary data table, and

(b) a second version identifier conveyed in a secondary data table and updated in response to at least one of,

a version change in said secondary table, and

a version change in a tertiary table hierarchically linked to said secondary table;

a processor for determining change in said secondary data table content by examining said second version identifier for a change following determination of a change in said first version identifier; and

an acquisition processor for acquiring said secondary data table in response to said determination of change.

2. (Original) Apparatus according to claim 1, wherein

said primary data table comprises a root database table for indicating version change in hierarchically ordered program guide data tables.

3. (Original) Apparatus according to claim 1, wherein

said secondary data table is used to indicate change in multimedia objects comprising objects associated with at least one of (a) broadcast channels, (b) broadcast programs, and (c) User interface controls.

4. (Original) Apparatus according to claim 1, wherein

said primary data table is used to indicate change in at least one of (a) electronic program guide information tables and (b) MPEG compatible program specific information.

5. (Original) Apparatus according to claim 1, wherein

said ancillary information is a two level hierarchical arrangement containing only a primary table and secondary tables.

6. (Original) Apparatus for adaptively decoding re-partitionable packetized program guide data, comprising:

a processor for acquiring program guide data comprising hierarchically ordered data table partitions and including partitioning information, said partitioning information including,

partition identifiers assigned to individual partitions of said program guide data, wherein said program guide data partitions are dynamically re-partitionable by re-assignment of said partition identifiers in said partitioning information; and

a processor for identifying said re-assigned partition identifiers and for acquiring additional program guide data in response to said identified re-assigned partition identifiers.

7. (Original) Apparatus according to claim 6, wherein

said partition identifiers identify program guide data partitions based on at least one of, (a) an area, (b) a broadcast time, (c) a complexity level, and (d) a partition type.

(claims continued on the next page)

8. (Original) A method for forming packetized program data to be suitable for processing in a decoder, comprising the steps of:

forming program guide information and ancillary information into hierarchically ordered data tables and including in said ancillary information,

(a) a first version identifier conveyed in a primary data table and updated in response to a version change in at least one of a plurality of secondary tables hierarchically linked to said primary data table, and

(b) a second version identifier conveyed in a secondary data table and updated in response to at least one of,

a version change in said secondary table, and

a version change in a tertiary table hierarchically linked to said secondary table; and

incorporating said ancillary information and said program guide information into packetized data for output to a transmission channel.

9. (Original) A method according to claim 8, including the step of

forming said primary data table to comprise a root database table for indicating version change in hierarchically ordered program guide data tables.

10. (Original) A method according to claim 8, wherein

forming said secondary data table to indicate change in multimedia objects comprising objects associated with at least one of (a) broadcast channels, (b) broadcast programs, and (c) User interface controls.

11. (Original) A method according to claim 8, wherein

forming said primary data table to indicate change in at least one of (a) electronic program guide information tables and (b) MPEG compatible program specific information.

12. (Original) A method according to claim 8, wherein

said ancillary information is a two level hierarchical arrangement containing only a primary table and secondary tables.

13. (Original) A method for forming packetized program data to be suitable for processing in a decoder, comprising the steps of:

partitioning program guide information and ancillary information into hierarchically ordered data table partitions and including a database in said ancillary information, said database including,

(a) updatable version numbers for indicating content change of a partition, and

(b) cell numbers assigned to individual partitions of said program guide information, wherein said program guide information cell partitions are dynamically re-partitionable by re-assignment of said cell number in said database; and

incorporating said ancillary information and said program guide information into packetized data for output to a transmission channel.

14. (Original) A method according to claim 13, wherein

said ancillary information contains a multimedia object comprising objects associated with at least one of (a) broadcast channels, (b) broadcast programs, and (c) User interface controls.

15. (Original) A method according to claim 14, wherein

an object comprises at least one of (a) a video segment, (b) an audio segment, (c) text, (d) an icon representing a user selectable item for display, (e) an HTML or SGML document (f) a menu of selectable items, (g) an image window for presentation within an encompassing image, and (h) an image window for initiating a multimedia function.

16. (Original) A method according to claim 13, wherein

a cell number incorporates at least one of, (a) an area identifier, (b) a broadcast time identifier, and (c) a complexity level identifier.

17. (Canceled).

18. (Canceled).

X. EVIDENCE APPENDIX (NONE)

None.

XI. RELATED PROCEEDINGS APPENDIX (NONE)

None.